



Title	Electric vehicle charging station placement: Formulation, complexity, and solutions
Author(s)	Lam, Albert Y S; Leung, Yiu Wing; Chu, Xiaowen
Citation	IEEE Transactions on Smart Grid, 2014, v. 5, n. 6, p. 2846-2856
Issued Date	2014
URL	http://hdl.handle.net/10722/219763
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Electric Vehicle Charging Station Placement: Formulation, Complexity, and Solutions

Albert Y.S. Lam, *Member, IEEE*, Yiu-Wing Leung, *Senior Member, IEEE*, and Xiaowen Chu, *Senior Member, IEEE*

Abstract—To enhance environmental sustainability, many countries will electrify their transportation systems in their future smart city plans, so the number of electric vehicles (EVs) running in a city will grow significantly. There are many ways to recharge EVs' batteries and charging stations will be considered as the main source of energy. The locations of charging stations are critical; they should not only be pervasive enough such that an EV anywhere can easily access a charging station within its driving range, but also widely spread so that EVs can cruise around the whole city upon being recharged. Based on these new perspectives, we formulate the EV charging station placement problem (EVCSP) in this paper. We prove that the problem is non-deterministic polynomial-time hard. We also propose four solution methods to tackle EVCSP, and evaluate their performance on various artificial and practical cases. As verified by the simulation results, the methods have their own characteristics and they are suitable for different situations depending on the requirements for solution quality, algorithmic efficiency, problem size, nature of the algorithm, and existence of system prerequisite.

Index Terms—Charging station, electric vehicle (EV), location, smart city planning.

NOMENCLATURE

G	Undirected graph modeling the city.
\mathcal{N}	Set of potential charging station construction sites.
\mathcal{E}	Set of connections connecting pairs of the construction sites.
n	Size of \mathcal{N} .
$d(i, j)$	Distance of the shortest path from nodes i to j .
f_i	Charging capacity of node i .
F_i	Demand requirement of node i .
D	Average traversable distance of fully charged electric vehicles.
\mathcal{N}'	Set of nodes with charging stations constructed.
α	A discount factor.
h_{ij}	Number of hops of the shortest path from nodes i to j in G .

x_i	Boolean variable for construction at node i .
x	Vector of x_i 's.
c_i	Construction cost at node i .
$\mathcal{N}_i^{\alpha D}$	Set of nodes within distance αD from node i .
\hat{G}	Induced graph from G .
$\hat{\mathcal{N}}$	Set of nodes in \hat{G} .
$\hat{\mathcal{E}}$	Set of edges in \hat{G} .
H	Induced subgraph of \hat{G} .
0^i	Source node of flow attached to node i .
x_0^i	Residue of flow remained in 0^i .
y_{jk}^i	Amount of flow on edge (j, k) originated from 0^i .
$N(H)$	Set of nodes associated to H .
C	A cost bound.
\tilde{G}	Undirected graph for the vertex cover problem.
$\tilde{\mathcal{N}}$	Set of nodes in \tilde{G} .
$\tilde{\mathcal{E}}$	Set of edges in \tilde{G} .
$\tilde{\mathcal{N}}$	Set of node for node selection in the greedy algorithm.

I. INTRODUCTION

DUE TO the world's shortage of fossil fuels, nations compete to secure enough reserves of natural resources for sustainability. Seeking alternative energy sources becomes crucial to a nation's future development. One of the major fossil fuel consumptions is transportation. Many daily heavily demanded vehicles are powered by gasoline. A major consequence of burning fossil fuels is the release of tremendous amount of harmful gases, which partially constitutes the global warming effect and deteriorates people's health. Electricity is considered as the most universal form of energy, which can be transformed from and to another form effectively. By converting the endurable renewable energy, like solar and wind energies, to electricity, we can manipulate energy in a much cleaner manner. Electrification of transportation, like deployment of electric vehicles (EVs), can not only alleviate our demand on fossil fuels, but also foster a better environment for living. Therefore, EVs will become the major components in the future transportation system.

EVs take the central role in this paper and they have been being studied actively since the boom of the smart grid. Incorporating EVs into an existing self-contained transportation system is challenging. Solely expanding the population of EVs in a city without enough road connections and corresponding charging and parking infrastructure will suppress the practicability of EVs due to their limiting moving ranges.

Manuscript received October 23, 2013; revised February 25, 2014 and June 8, 2014; accepted July 22, 2014. Date of publication August 15, 2014; date of current version October 17, 2014. The work of A. Y. S. Lam was supported by the Faculty Research Grant of Hong Kong Baptist University under Grant FRG2/13-14/045. A preliminary version of this paper was presented at the IEEE International Conference on Smart Grid Communications, Vancouver, BC, Canada, October 2013. Paper no. TSG-00800-2013.

The authors are with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (e-mail: ayslam@comp.hkbu.edu.hk; ywleung@comp.hkbu.edu.hk; chxw@comp.hkbu.edu.hk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSG.2014.2344684

Moreover, existing gas stations are primarily designed for gas refueling; combining charging infrastructure with the conventional gas stations may not be appropriate as the relatively longer charging process will saturate the limited space of the gas stations. We need to carefully plan EV charging facilities to modernize the transportation system. To be precise, we study how EVs will be integrated into the transportation system seamlessly with a focus on charging stations and this will help make our cities “smart.”

We study the EV charging station placement problem (EVCSP) by finding the best locations to construct charging stations in a city. An EV should always be able to access a charging station within its capacity anywhere in the city. Charging stations should be built widely enough such that the moving range of an EV can be extended to every corner of the city by having the EV recharged at a charging station available nearby. We study the locations where charging stations should be constructed in a city such that we can minimize the construction cost with coverage extended to the whole city and fulfillment of drivers’ convenience. In this paper, we formulate the problem as an optimization problem, based on the charging station accessibility and coverage in the city. We also study its complexity and propose various methods to solve the problem.

In this paper, we focus on the long-term human aspects rather than the technological ones. The smart city plan and technology advancement take different time-spans for realization. To meet the government policy in some countries, the population of EVs needs to be boosted. Satisfaction and convenience of drivers have strong impacts on the growth of EVs in a city. The ease of recharging their EVs is one of the most important considering factors when one decides to buy an EV [2]. Population density and the demand for charging facilities for their EVs are passive factors. The influence of these human factors usually takes longer (say 5–10 years) to be realized. On the other hand, technology advances in a much faster pace and the impact of the charging loads to the grid will be lessened with practical technological solutions, especially for security and reliability issues.

As a whole, the complete charging station problem with consideration of all possible considering factors can be framed as a two-level problem. In the first level, a set of potential locations for charging station constructions can be determined based on some urban planning factors, e.g., land use type, environment impact, and safety, and also some engineering factors addressed in some of the previous work explained in the next section. In the second level, charging station placement is further enhanced from the drivers’ perspective and we place charging stations in the potential locations determined from the first level. So this paper mainly falls into the second level. This arrangement allows us to focus on examining the problem from a new angle. Furthermore, a model without too much technical details of the charging facilities allows us to retain flexibility for different charging technologies and standards. For example, charging with connected power cables can be replaced by battery swapping. Our model can still be applied to the scenario with battery swapping EVs. We focus on the human factors and it can be served as the foundation for various kinds of charging specifications.

Our main contributions include formulating the new problem EVCSP, analyzing its complexity, and proposing several solutions to the problem. The rest of this paper is organized as follows. Related work is given in Section II. We formulate the problem in Section III and discuss its complexity in Section IV. Section V presents four solution methods. In Section VI, simulation results are provided for performance evaluation and we also compare the solution methods in terms of characteristics and suitability for different situations. Finally, we conclude this paper in Section VII.

II. RELATED WORK

Most of the existing work on EVs is related to studying the operational influence of EVs on the grid, i.e., how power is transferred from and to the grid. Besides charging scheduling [3], in a vehicle-to-grid (V2G) system, hundreds of EVs are coordinated to act as a power source selling power back to the grid or to support auxiliary services, like regulation. A multilayer market for V2G energy trading was proposed in [4]. The market price was settled via double auction and the proposed mechanism could maximize the EVs’ revenues. In [5], a queueing network was utilized to model the dynamics of EVs participating in V2G. The model could facilitate service contract engagement for regulation ancillary services. Yu *et al.* [6] investigated the joint scheduling of EVs and unit commitment and this allowed us to optimize the system’s total running cost with the presence of EVs. Guo *et al.* [7] discussed the incorporation of PV equipment into charging stations. It considered that charging facilities equipped with PV panels and the stored solar energy, together with the power requested from the grid, can be used to power EVs. Etezadi-Amoli *et al.* [8] and Masoum *et al.* [9] studied the impact of EV charging to the performance of power distribution networks with the presence of charging stations, which can represent rapid heavy loads. Etezadi-Amoli *et al.* [8] illustrated the effect of fast-charging EVs in terms of power-flow, short-circuit, and protection while Masoum *et al.* [9] proposed a new smart load management strategy to coordinate EVs for peak load shaving, power loss minimization, and voltage profile improvement. However, this paper is dedicated to studying the locations for building charging stations, which is an important aspect of the smart city plan.

References [10] and [11] investigated the location and sizing issues of charging stations; Liu *et al.* [10] handled the two issues separately while Liu *et al.* [11] considered a joint optimization for both. In consideration of environmental factors (e.g., load locations, load balance, power quality, etc.) and service radius of charging stations, candidate sites in [10] were selected with a two-step screening method instead of optimization. Liu *et al.* [11] constructed an optimization problem in which various kinds of costs (including construction, operating, and charging costs) were minimized with traffic flow and charging requirement constraints and particle swarm optimization heuristic was adopted to compute the solution of the nonconvex problem. Jia *et al.* [12] studied the siting and sizing issues, where the locations and numbers of chargers at each

site are determined at the same time with consideration of charging demand. Pashajavid and Golkar [13] discussed how to allocate charging stations with the presence of solar generation. Chen *et al.* [14] determined the charging station locations with real-world public parking information of Seattle as inputs. It formed a mixed-integer program (MIP) by minimizing the total access costs to drivers' destinations from charging stations. Wang *et al.* [15] discussed the design of power architectures and power electronics circuit topologies for high power superfast EV charging stations with enhanced grid support functionality. Another related problem is the gas station problem described in [16]. However, it was not related to gas station placement but determined the cheapest route connecting gas stations with other locations. In operations research, the study of placing facilities, such as gas stations and fire stations, is generally cast as facility location problems [17], e.g., the maximal covering location problem [18]. It concerned about the distances or times to travel to individual facilities from various locations. Such a model cannot guarantee that the induced subgraph constituted by the facility locations is connected but this condition is significant in our charging station placement model. Moreover, Mayfield [19] gave a general discussion about the interior design of charging stations in various parking facility types instead of analyzing in the engineering perspective. However, in this paper, we focus on the long-term issues of charging station placement for smart city planning, where the short-term factors (e.g., instantaneous loads) will be of relatively less importance. To the best of our knowledge, we are the first to study charging station placement from the new perspectives of the drivers' convenience and EVs' accessibility. Other factors, like traffic conditions, may also be taken into account but they are out of the scope of this paper.

III. PROBLEM FORMULATION

A. System Model

We model a city with an undirected graph $G = (\mathcal{N}, \mathcal{E})$, where \mathcal{N} and \mathcal{E} denote the sets of possible sites for constructing charging stations and connections between pairs of sites, respectively. Suppose $|\mathcal{N}| = n$. Let $d : \mathcal{N} \times \mathcal{N} \rightarrow \mathbb{R}^+$ be the distance function, where $d(i, j)$ denotes the distance of the shortest path from nodes i to j by traversing the connections.¹ Let f_i be the capacity of node i representing the average capacity of charging service supported if a charging station is constructed at location i . It is related to the size of the site and traffic conditions in the surrounding. Each node i also has a demand requirement F_i , which refers to its average local charging demand. The more EVs are based at location i , the higher F_i is. F_i can be estimated from the population size and the EV penetration rate of that location. Without loss of generality, we assume that some F_i 's are positive while some are of zero value.

We define D to be the average distance able to be traversed by most typical EVs available in the market when being fully

charged. A subset of nodes $\mathcal{N}' \subset \mathcal{N}$ is said to be reachable by D if the following conditions hold.

- 1) For each $i \in \mathcal{N}'$, there exists a node $j \in \mathcal{N}'$ such that $d(i, j) \leq D$.
- 2) For each $i \in \mathcal{N}$, the total capacity, constituted from those nodes $j \in \mathcal{N}'$ such that $d(i, j) \leq \alpha D$ with discount factor $\alpha \in (0, 1]$, is greater than or equal to F_i .
- 3) For any $i, j \in \mathcal{N}'$, suppose h_{ij} be the number of hops of the shortest path from i to j in G . The distance of the path $d(i, j)$ should be smaller than or equal to $h_{ij}D$.

\mathcal{N}' represents the set of locations which have been selected with charging stations constructed. A city is well planned if \mathcal{N}' is reachable by D . With Condition 1, an EV, which has been fully charged at one location, can recharge at another site within distance D away. Condition 1 guarantees that EVs will not be confined in one single location (or area). Condition 2 says that the local charging demand at a location (e.g., F_i at node i) must be satisfied by the total charging capacities contributed by those charging stations located within distance αD away. α is used to model the tolerance of drivers to move away from their current locations for recharging. Its maximum value is one because an EV can traverse for a distance at most D . The smaller α , the more conservative the model is, i.e., more charging stations should be placed around every possible location. With Condition 3, the charging station network, where each charging station is separated with another of at most distance D , spans the whole city. Note that, we use one single D to characterize the accessibility of the whole city for all kinds of EV models because the distribution of the charging stations should cater for all possible EVs traveling on the roads. To do this, we should assign D with a more conservative value, e.g., the maximum travel distance of the most basic EV model in the market when being fully charged. To summarize, the conditions all together guarantee that the serving areas of the charging stations cover every corner of the city for all possible EVs.

B. Formulation

Let x_i be the decision (Boolean) variable indicating if node i is chosen for placement and c_i be its construction cost. We minimize the total cost as the objective, i.e., $\sum_{i=1}^n c_i x_i$.

For each i , we define $\mathcal{N}_i^{\alpha D} = \{j \in \mathcal{N} | d(i, j) \leq \alpha D\}$, representing the set of nodes (including node i itself) within distance αD from i . We can restate Condition 2 as $\sum_{j \in \mathcal{N}_i^{\alpha D}} f_j x_j \geq F_i, \forall i \in \mathcal{N}$. As Condition 3 holds for any pair of nodes, Condition 3 implies Condition 1. To restate Condition 3, we first create a graph $\hat{G} = (\hat{\mathcal{N}}, \hat{\mathcal{E}})$, where $\hat{\mathcal{N}}$ is set to \mathcal{N} and $\hat{\mathcal{E}}$ is equal to $\{(i, j) | i, j \in \mathcal{N}, d(i, j) \leq D, i \neq j\}$ (see the example shown in [1, Fig. 1]). Consider those nodes i in G with $x_i = 1$ (i.e., \mathcal{N}') and they constitute the corresponding induced subgraph H of \hat{G} . Condition 3 is equivalent to having H connected. In other words, H has one single connected component. Instead of inspecting the original graph G , we can focus on \hat{G} to formulate the problem. Similar to [20], we adopt a network flow model to address Condition 3. Consider that there is some virtual flow² flown from some sources to some sinks. If the sources and the sinks are not connected, the flow

¹The distance $d(i, j)$ refers to the distance of an actual path connecting locations i and j but not the Euclidean distance.

²The virtual flow here is independent of the traffic flow.

from the sources cannot reach the sinks. Imagine that there is a source node 0^i attached to node i and it has n units of flow available to be sent along \hat{G} through node i . Let $0 \leq x_0^i \leq n$ be the residue of flow not consumed by the network. Each node j with $x_j = 1$ will consume one unit of flow. For each edge $(j, k) \in \hat{\mathcal{E}}$, we indicate the amount of flow on (j, k) originated from 0^i with variable y_{jk}^i . Hence, we can guarantee that the flow can reach those nodes j with $x_j = 1$ from node i on \hat{G} with the following:

$$x_0^i + y_{0i}^i = n \quad (1)$$

$$0 \leq y_{jk}^i \leq nx_k, \forall (j, k) \in \hat{\mathcal{E}} \cup (0^i, i) \quad (2)$$

$$\sum_{j|(j,k) \in \hat{\mathcal{E}}} y_{jk}^i = x_k + \sum_{l|(k,l) \in \hat{\mathcal{E}}} y_{kl}^i, \forall k \in \hat{\mathcal{N}} \quad (3)$$

$$\sum_{j \in \hat{\mathcal{N}}} x_j = y_{0i}^i \quad (4)$$

$$0 \leq x_0^i. \quad (5)$$

Equation (1) says that the total amount of flow y_{0i}^i going out of the source 0^i and the retained x_0^i in 0^i is n , where n is the number of nodes in G and it is the upper bound of flow possible to be absorbed in the network. Equation (2) confines that only a sink can receive incoming flow and (3) describes that the total incoming flow to a node is equal to the total outgoing flow plus the amount for a sink. Equation (4) explains that the total flow getting out of the source is equal to the total absorbed by the sinks and (5) restricts nonnegative residue remained in the source. An illustrative example of the network flow model is given in the Appendix.

Note that (1)–(5) require node i to be selected for charging station construction. Otherwise, no flow from source 0^i is allowed to be delivered to the sinks. To cater for this requirement, we attach a source node to each node in $\hat{\mathcal{N}}$ and the overall mathematical formulation of EVCSPP is modified accordingly as follows:

$$\text{minimize } \sum_{i=1}^n c_i x_i \quad (6a)$$

$$\text{subject to } \sum_{j \in \mathcal{N}_i^{\alpha D}} f_j x_j \geq F_i, \forall i \quad (6b)$$

$$x_i = \{0, 1\}, \forall i \quad (6c)$$

$$x_0^i + y_{0i}^i = n, \forall i \in \hat{\mathcal{N}} \quad (6d)$$

$$0 \leq y_{jk}^i \leq nx_i x_k, \forall (j, k) \in \hat{\mathcal{E}} \cup (0^i, i), \forall i \in \hat{\mathcal{N}} \quad (6e)$$

$$\sum_{j|(j,k) \in \hat{\mathcal{E}}} y_{jk}^i = x_i x_k + \sum_{l|(k,l) \in \hat{\mathcal{E}}} y_{kl}^i, \forall i, k \in \hat{\mathcal{N}} \quad (6f)$$

$$x_i \sum_{j \in \hat{\mathcal{N}}} x_j = y_{0i}^i, \forall i \in \hat{\mathcal{N}} \quad (6g)$$

$$0 \leq x_0^i, \forall i \in \hat{\mathcal{N}}. \quad (6h)$$

Equations (6a) and (6b) have been discussed before. Equation (6c) confines x_i to be a Boolean variable. Equations (6d)–(6h) correspond to the induced connected subgraph condition [i.e., (1)–(5)] for all nodes.

Equation (6) is an MIP with Boolean variables x_i 's and continuous variables y_{jk}^i 's. With the quadratic terms in equality

constraints (6f) and (6g), it is not a mixed-integer linear program (MILP). Hence, this problem is not easy to be solved.

Before examining the complexity of the problem, we discuss the relationship of our model with the grid. The range of power demand from a charging station can be inferred from the scale of the charging station (in terms of the number of chargers) and the usage pattern. With this information, the utility company which manages the distribution network can assess the risk of potential security problems from the expected loads of the charging stations. In the initial set of potential locations, we only consider those feasible places which allow power facility upgrade. We can make use of the charging capacity defined in constraint (6b) to model this. Moreover, many practical solutions for security and reliability are available to be incorporated into the grid easily. The charging stations can also be equipped with energy storage and renewable energy generation [e.g., from solar photovoltaic (PV) setup]. In addition, practical methods, like installation of electric spring [21] and distributed active and reactive power injection control [22], can be easily adopted to regulate the voltage with fast response time. Hence, the impact of sudden large energy demand leading to high voltage drop can be alleviated.

IV. COMPLEXITY ANALYSIS

The decision version of EVCSPP can be framed as follows. Let $N(H)$ be the set of nodes associated to the induced subgraph H . Each node i has a capacity $f_i \in \mathbb{Z}^+$ and a demand F_i and it is associated with the node set $\mathcal{N}_i^{\alpha D}$. Given an undirected graph $\hat{G} = (\hat{\mathcal{N}}, \hat{\mathcal{E}})$, with the cost $c_i \in \mathbb{Z}^+, \forall i$, and a cost bound $C \in \mathbb{Z}^+$, does there exist an induced subgraph H of \hat{G} such that: 1) for each $i \in \hat{\mathcal{N}}$, $\sum_{j \in \mathcal{N}_i^{\alpha D} \cap N(H)} f_j \geq F_i$; 2) H is connected; and 3) $\sum_{i \in N(H)} c_i \leq C$?

Theorem 1: The decision version of EVCSPP is nondeterministic polynomial-time (NP)-complete.³

Proof: Similar to [20], we construct a reduction from the vertex cover problem (VCP) to EVCSPP. In the graph $\tilde{G} = (\tilde{\mathcal{N}}, \tilde{\mathcal{E}})$, a vertex cover is a subset of nodes $\mathcal{N}' \subset \tilde{\mathcal{N}}$ such that each edge $(i, j) \in \tilde{\mathcal{E}}$ has either i or j , or both in \mathcal{N}' . Without loss of generality, we assume $\tilde{\mathcal{E}} \neq \emptyset$. VCP determines if there exists a vertex cover \mathcal{N}' of \tilde{G} with $|\mathcal{N}'| \leq C$.

We create a graph $\hat{G} = (\hat{\mathcal{N}}, \hat{\mathcal{E}})$, where $\hat{\mathcal{N}} = \tilde{\mathcal{N}} \cup \tilde{\mathcal{E}}$ and $\hat{\mathcal{E}}$ is constructed as follows. For each pair of distinct nodes $i, j \in \tilde{\mathcal{N}}$, we create an edge (i, j) in $\hat{\mathcal{E}}$; for each $e = (i, j) \in \tilde{\mathcal{E}}$, we append (i, e) and (e, j) to $\hat{\mathcal{E}}$. For each $i \in \tilde{\mathcal{N}}$, its cost is set as $c_i = 1$ and zero otherwise. For each $e \in \tilde{\mathcal{E}}$, we set $f_e = 1$ and zero otherwise. We also set $\mathcal{N}_i^{\alpha D} = \tilde{\mathcal{E}}$ and $F_i = |\tilde{\mathcal{E}}|$ for all $i \in \tilde{\mathcal{N}}$.

We claim that VCP on \tilde{G} has a cost upper bound C if and only if EVCSPP has a solution with cost at most C . Let \mathcal{N}' be a vertex cover of \tilde{G} with $|\mathcal{N}'| \leq C$ and H be the induced subgraph of \hat{G} by nodes $\mathcal{N}' \cup \tilde{\mathcal{E}}$. It is easy to verify that $|\mathcal{N}_i^{\alpha D} \cap N(H)| = |\tilde{\mathcal{E}}|$ and thus $\sum_{j \in \mathcal{N}_i^{\alpha D} \cap N(H)} f_j = |\tilde{\mathcal{E}}| = F_i$. As \mathcal{N}' is a vertex cover, each $e = (i, j) \in \tilde{\mathcal{E}}$ must have at least

³In computational complexity theory, a decision problem is NP-complete if it is in the intersection of NP and NP-hard problem sets [23]. There is no known method to solve such problem in polynomial time.

one of i and j in \tilde{N} and thus H must contain an edge (e, k) for some $k \in \mathcal{N}'$. Moreover, \tilde{N} forms a clique in \hat{G} . Hence, H must be connected. Since each $e \in \tilde{\mathcal{E}} \subset \tilde{N}$ imposes no cost, H has the same cost as \mathcal{N}' in \hat{G} . Therefore, EVCSPP has a solution with cost at most C .

Consider that an induced subgraph H is a solution of EVCSPP. We set $\mathcal{N}' = N(H) \cap \tilde{N}$. H contains $\tilde{\mathcal{E}}$: As $f_j = 1$ for $j \in \tilde{\mathcal{E}}$, for any $i \in \hat{N}$, $F_i = |\tilde{\mathcal{E}}|$ guarantees $\tilde{\mathcal{E}} \subset N(H)$. Since H is connected, each $i \in \mathcal{N}'$ must have an edge with an $e \in \tilde{\mathcal{E}}$ in \hat{G} . Moreover, \mathcal{N}' has at most C nodes. Hence, \mathcal{N}' is a vertex cover of \tilde{G} with $|\mathcal{N}'| \leq C$. ■

With Theorem 1, we have an immediate corollary as follows.

Corollary 1: EVCSPP is NP-hard.⁴

V. PROPOSED SOLUTIONS

Since the problem is NP-hard, there is no trivial way to solve it. In this section, we propose four solution methods to tackle the problem. They possess their own pros and cons and one method may be more suitable for a particular situation than another.

A. Method I: Iterative MILP

Equations (1)–(5) can be used to guarantee that the solution subgraph constituted by all nodes j with $x_j = 1$ is connected as long as $x_i = 1$. If we assume that node i is one of locations for charging station construction, i.e., $x_i = 1$, Equation (6) can be reduced to

$$\text{minimize } \sum_{k=1}^n c_k x_k \quad (7a)$$

$$\text{subject to } \sum_{j \in \mathcal{N}_k^{gD}} f_j x_j \geq F_k, \forall k \in \hat{N} \quad (7b)$$

$$x_k = \{0, 1\}, \forall k \in \hat{N} \quad (7c)$$

$$x_0^i + y_{0i}^i = n \quad (7d)$$

$$0 \leq y_{jk}^i \leq n x_k, \forall (j, k) \in \hat{\mathcal{E}} \cup (0^i, i) \quad (7e)$$

$$\sum_{j|(j,k) \in \hat{\mathcal{E}}} y_{jk}^i = x_k + \sum_{l|(k,l) \in \hat{\mathcal{E}}} y_{kl}^i, \forall k \in \hat{N} \quad (7f)$$

$$\sum_{j \in \hat{N}} x_j = y_{0i}^i \quad (7g)$$

$$0 \leq x_0^i \quad (7h)$$

$$x_i = 1. \quad (7i)$$

Equation (7) is an MILP and it can be solved with standard MIP solvers applying methods like branch-and-bound. Now the question becomes which node i should be chosen for this purpose.

We write the solution of (7) as $\inf_{x \in \Omega_i} \sum_{j=1}^n c_j x_j$, where Ω_i is the solution space constituted by (7b)–(7i) and \inf refers to the infimum operator. We have the following theorem.

Theorem 2: The solution of (6) can be determined by solving

$$\min \left(\inf_{1 \leq i \leq n} \sum_{j=1}^n c_j x_j \right). \quad (8)$$

⁴An optimization problem is NP-hard if it is at least as hard as the hardest problems in the NP problem set [23].

Proof: The solution of (6) can induce a connected subgraph of \hat{G} where (6b) is satisfied and the total cost function (6a) is minimized. When we solve (7), its solution induces a connected subgraph of \hat{G} with minimum total cost satisfying (6b) where node i is included. To compute (8), we apply (7) to every node i (thus we solve (7) n times) and its solution is the minimum among the n subgraphs. Since (6a) and (6b) exist in every (7), the solution of (6) must be one of the computed subgraphs with some node i included. Hence, addressing (8) can solve (6). ■

With this result, the original mixed-integer nonlinear program becomes n solvable MILPs. Since we need to go through all nodes iteratively, we call this method iterative MILP.

Besides the fact that the computational time of solving MILP (7) grows super-linearly with n , this method also suffers from the problem that the number of MILPs [i.e., (7)] needed to be solved also increases with n . Hence, the combined effect of increasing n will make its computation time accelerating extraordinarily fast. Hence, this method is only applicable to small problem instances. If the solver applied to (7) can produce the optimal solution, Method I will guarantee the optimality.

B. Method II: Greedy Approach

Here, we present an efficient greedy algorithm, which is applicable to the original formulation (6) and requires much shorter computation time. Before discussing its details, we have the following lemma to facilitate its development.

Lemma 1: Equations (6) and (7) is feasible if and only if $x = [x_1, \dots, x_n] = [1, \dots, 1]$ is a feasible solution, which gives an upper bound of the objective function value $\sum_{i=1}^n c_i$.

Proof: First, we consider the only if-direction. As the problem is feasible, there exists a feasible $x' = [x'_1, \dots, x'_n]$, composed of some 0s and/or 1s, satisfying constraints (6b)–(6h). If $x'_i = 1$ for all i , then we have the result. Consider that there is at least one j such that $x'_j = 0$. If we produce another x'' by modifying x'_j with value one, besides (6c), x'' will always satisfy constraint (6b), as we will not change or increase the sum on the left-hand side of (6b). Moreover, as $0 < \alpha \leq 1$, if $x'_j = 0$ satisfies (6b), there exists at least one node k with $x'_k = x''_k = 1$ within distance D away from node j . In this way, if we have $x'_j = 1$, we will attach node j to the subgraph induced by x' through node k . In other words, the subgraph induced by x'' is still connected, i.e., satisfying (6d)–(6h). We can repeat this process until we change all 0s to 1s and this produces x with upper bound $\sum_{i=1}^n c_i$.

The if-direction is trivial. We complete the proof. ■

Corollary 2: If $x = [x_1, \dots, x_n] = [1, \dots, 1]$ is not feasible, EVCSPP is infeasible.

Corollary 2 can be used to check the feasibility of a problem instance.

Assume that we have a feasible problem instance. We construct a greedy algorithm by reducing the total cost as much as possible in each iteration and it results in a sub-optimal solution. Its pseudocode is given in Algorithm 1. In Line 1, we start with the feasible $x = [x_1, \dots, x_n] = [1, \dots, 1]$ explained in Lemma 1 and then go through a certain number of iterations

Algorithm 1 Greedy Algorithm

```

1: Set  $x_i = 1$  for  $i = 1, \dots, n$ .
2: repeat
3:   Construct a node set  $\bar{\mathcal{N}}$  composed of nodes  $i$  with  $x_i = 1$  where the
     induced subgraph is still connected when  $x_i$  is set to zero.
4:    $x' \leftarrow x$ 
5:    $flag \leftarrow 0$ 
6:   repeat
7:     Select  $j$  with the largest  $c_j$  in  $\bar{\mathcal{N}}$ .
8:     Modify  $x'$  by setting  $x'_j = 0$ 
9:     if  $x'$  satisfies (6b) then
10:       $x \leftarrow x'$ 
11:       $flag \leftarrow 1$ 
12:     else
13:       $x'_j \leftarrow 1$ 
14:      Remove  $j$  from  $\bar{\mathcal{N}}$ 
15:     end if
16:   until  $flag = 1$  OR  $\bar{\mathcal{N}} = \emptyset$ 
17: until  $\bar{\mathcal{N}} = \emptyset$ 

```

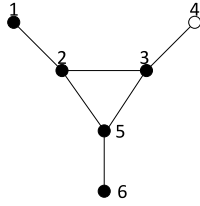


Fig. 1. Node selection in the greedy algorithm.

(Lines 2–17). In each iteration, we select those nodes in the subgraph induced by x which will not disconnect the subgraph if we remove them from the subgraph and we call this selection $\bar{\mathcal{N}}$ (Line 3). For example, Fig. 1 shows a graph \hat{G} of six nodes, where a dot i and a hole j mean $x_i = 1$ and $x_j = 0$, respectively. In this case, we have $\bar{\mathcal{N}} = \{1, 3, 6\}$. We can see that the resultant x' formed by removing any one node in $\bar{\mathcal{N}}$ will still satisfy Constraints (6d)–(6h). Then, we attempt to deselect the one (say node j) with the highest cost c_j in $\bar{\mathcal{N}}$ (Line 7). If the resultant x' satisfies (6b), x' is a feasible point and we proceed to the next iteration (Lines 9–11). Otherwise, we remove j from $\bar{\mathcal{N}}$ (Line 14). Instead of deselecting j (Line 13), we deselect the one with the next highest cost. The iterations terminate when no nodes remain in $\bar{\mathcal{N}}$ (Line 17). The final solution x is the best determined by the greedy algorithm. Note that the resultant solution is usually sub-optimal, especially when the problem size n becomes larger.

C. Method III: Effective MILP

Recall that in Method I, we need to apply MILP (7) to every node since we are not sure which node i has $x_i = 1$ in the optimal solution and thus it is usually subject to long computation time. However, if we know the node which has unity in the solution, we can save lots of effort by applying (7) to that node only. With Theorem 3, under a general condition, we find that not all nodes i are required to generate the solution as in Method I.

Theorem 3: Suppose that the demand requirements F_i for all i are positive. Then for any node i , at least one node j in i 's one-hop neighborhood in \hat{G} , i.e., $\mathcal{N}_i^{\alpha D}$, must have $x_j = 1$ in the optimal solution of (6).

Proof: Since all F_i 's are positive, $[x_1, \dots, x_n] = [0, \dots, 0]$ can never be a solution. Hence, the solution must contain at least one node i with $x_i = 1$. Consider any particular node i . If x_i is set to one in the optimal solution, then the result holds by assigning j to i as $i \in \mathcal{N}_i^{\alpha D}$.

Consider that x_i is set to zero. As F_i is positive, at least one term $f_j x_j$ on the left-hand side of (6b) must be positive. In other words, at least one node (e.g., j) in $\mathcal{N}_i^{\alpha D}$ has $x_j = 1$. Since $\alpha \leq 1$, node j must be a one-hop neighbor of node i . Since all F_i 's are positive, the condition applies to every node.

Hence, the result is true for every node in the graph. ■

With this theorem, we compose Method III by choosing any node i and applying (7) with respect to those nodes in $\mathcal{N}_i^{\alpha D}$ only. The number of (7) required to be solved depends on the cardinality of $\mathcal{N}_i^{\alpha D}$. We can minimize the computational time by choosing the node i with the smallest degree in \hat{G} . In this way, we can simplify Method I by exploiting the network structure of the graph and the solutions of both Methods I and III are equivalent. Similar to Method I, If the solver applied to (7) can produce the optimal solution, Method III will also guarantee the optimality. Since EVs are movable in a city, it is common to have EVs appearing in every location (node) in a certain time-span, and thus we have positive F_i for all nodes i . Hence, the condition imposed in Theorem 3 generally holds in most situations.

D. Method IV: Chemical Reaction Optimization

Chemical reaction optimization (CRO) is a recently proposed nature-inspired metaheuristic for optimization [24]. Under certain conditions, it has been proved to be able to converge to the global optimum for combinatorial optimization problems (like EVCSPP) [25] and it has been demonstrated to have very good performance in solving real-world problems, e.g., [26] and [27]. CRO is general-purpose and we apply CRO to EVCSPP. In CRO, the manipulated agents are molecules, each of which carries a solution. The molecules explore the solution space of the problem through a random series of elementary reactions taking place in a container. We define four types of elementary reactions, each of which has its own way to modify the solutions carried by the involved molecules. Due to space limitation, we do not illustrate every detail of CRO but explain the necessary modifications based on the framework described in [24]. We basically follow [24] to construct the algorithm. It consists of four elementary reactions, including on-wall ineffective collision, decomposition, intermolecular ineffective collision, and synthesis. They are implemented as follows.

1) *On-Wall Ineffective Collision:* It mimics that a molecule hits a wall of the *Container* and then bounces back. This elementary reaction is not vigorous and we only have small modifications to the molecule. Let x and x' be the solutions held by the molecules before and after the change. We apply our greedy approach (Method II)⁵ to x to produce x' , i.e., $x \xrightarrow{\text{greedy}} x'$.

⁵Note that we can initiate the greedy algorithm with any x instead of the unity vector $[1, \dots, 1]$ by skipping Line 1 in Algorithm 1.

TABLE I
SIMULATION RESULTS FOR $n = 50$ AND $D = 20$ KM

α	Matched/ Feasible cases	Objective function value (No. of stations constructed)						Computation time (s)			
		Method I	Method II	Method III	Method IV (mean)	Method IV (best)	Method IV (worst)	Method I	Method II	Method III	Method IV (mean)
1	32/100	9.4875 (24.7500)	9.7553 (25.7500)	9.4875 (24.7500)	9.6119 (25.2150)	9.5092 (24.8800)	9.7158 (25.5700)	478.6861	0.0610	23.6505	20.4516
0.9	20/62	10.8950 (27.2097)	11.0986 (28.0000)	10.8950 (27.2097)	11.0050 (27.5435)	10.9235 (27.2903)	11.0745 (27.8387)	369.5348	0.0618	18.2473	23.3749
0.8	8/23	12.5701 (30.4783)	12.8035 (31.2174)	12.5701 (30.4783)	12.6786 (30.6826)	12.5818 (30.4783)	12.7829 (30.9130)	310.1252	0.0611	15.7985	29.6049
0.7	0/2	13.3116 (33.5000)	13.8089 (34.5000)	13.3116 (33.5000)	13.4800 (33.5000)	13.3182 (33.5000)	13.7777 (33.5000)	331.2596	0.0542	19.9461	34.9514

2) *Decomposition*: It describes that one molecule hits a wall and breaks into two separate molecules. It involves vigorous changes to the molecules. Let x be the solution held by the reactant molecule and x'_1 and x'_2 be the solutions of the resultant molecules. Here, x'_1 and x'_2 are randomly generated in the solution space. A random solution can be produced by modifying Algorithm 1 where the repeat loop (Lines 6–16) iterates for a random number of times (between 1 to n) and we select a random node in \bar{N} in Line 7.⁶ A decomposition can be described as $x \xrightarrow{\text{random}} x'_1 + x'_2$.

3) *Intermolecular Ineffective Collision*: It portrays that two molecules collide with each other and then bounce away. The change is not vigorous. Let x_1 and x_2 be the two reactant molecules and x'_1 and x'_2 be the resultant molecules. Similar to the on-wall ineffective collision, we apply the greedy algorithm to the respective molecules to modify the solutions, i.e., $x_1 + x_2 \xrightarrow{\text{greedy}} x'_1 + x'_2$.

4) *Synthesis*: Synthesis describes that two molecules collide with each other and then combine into one molecule. The change is vigorous. Similar to decomposition, we produce a new molecule by randomly generating its solution in the solution space. Let x_1 and x_2 be the two reactant molecules and x' be the resultant molecule. We have $x_1 + x_2 \xrightarrow{\text{random}} x'$.

When initializing the algorithm, we assign random solutions in the solution space to the molecules (this can be done by the random solution generation used in decomposition). It is clear that Method II is embedded in this method except that Method II always start with a unity vector x . We can guarantee that Method IV is always superior to Method II in terms of solution quality by having at least one molecule possessing a unity vector as its initial solution. So we can assign the unity vector to some initial molecules (say 10%) in the initialization phase. Since CRO is a probabilistic algorithm, the solutions produced in different runs could be different.

VI. PERFORMANCE STUDY

A. Simulation Results

We perform a series of simulations to evaluate the performance of the four solution methods. All simulations are run on the same computer with Intel Core i7-3770 CPU at 3.40 GHz and 16 GB of RAM, and conducted in the MATLAB environment. For Methods I and III, the MILP is computed

with the CPLEX solver [28] and YALMIP [29]. Recall that Methods I, II, and III are deterministic while Method IV is probabilistic. For illustrative purposes, we repeat Method IV ten times for each simulation case. After several trial runs, we set the parameter values for Method IV as: “function evaluation (FE) limit” = 2000, “initial kinetic energy (KE)” = 10, “initial population size” = 40, “initial buffer” = 10, “collision ratio” = 0.5, “synthesis threshold” = 0.5, “decomposition threshold” = 20, and “KE loss rate” = 0.9. We conduct three tests. In the first test, we examine the solutions’ performance with changing α . The second aims to study how the computation time grows with the problem size. In the third, we test how the solution methods perform in a real-world scenario.

In the first test, we randomly generate 100 feasible instances. Each instance of G is constructed by randomly placing 50 nodes in an area of 100×100 km², where we assign a random value in the range of (0, 1] to the cost c_i , and D , f_i and F_i , for all i , are set to 20 km, 0.5, and 1, respectively. For simplicity, we assume that the nodes are interconnected and the length of the shortest path of each pair of nodes is determined with the Euclidean distance between them. As explained in Section III, we can produce \hat{G} from G . Then, we can check the feasibility of each instance with Corollary 2.

We verify the performance of the four methods with respect to the computed solution quality and the computation time. The results are given in Table I. The second column indicates the number of feasible and matched cases among the 100 graphs. All graphs are feasible when α is equal to one. When α decreases, the number of resulted feasible cases will also decrease as constraint (6b) becomes stronger. The matched cases indicate those of the feasible ones producing the same objective function values by all the four approaches. Regardless of the nonstatistically significant cases with $\alpha = 0.7$, all the four methods can produce the best solutions for around 1/3 of the feasible cases. The other columns show the average objective function values and computation times of the four methods for the feasible cases. For Method IV, we also provide the average (among the cases) of the best (among the repeats) and the worst (among the repeats) for reference. The average numbers of charging stations appeared in the solutions are put in brackets. Methods I and III always give the best solutions and Method IV always outperforms Method II. In terms of computation time, Method II is the fastest and Method III comes the second. Method IV is the next and Method I takes the longest.

⁶Although the generations of x'_1 and x'_2 do not rely on x , the energies stored in the molecules do. Interested readers may refer to [24] for more information.

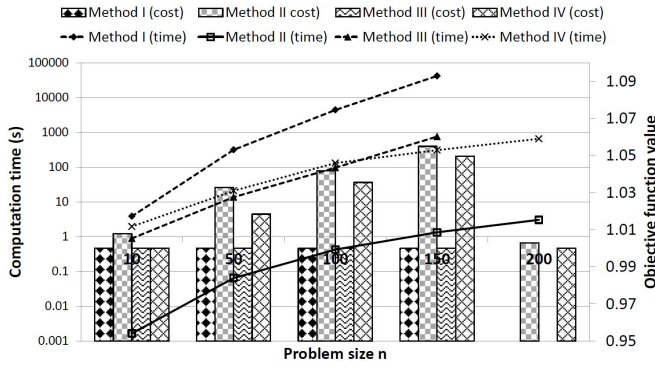


Fig. 2. Computation time changing with problem size.

TABLE II
AVERAGE NUMBER OF STATIONS COMPUTED IN THE SECOND TEST

	Method I	Method II	Method III	Method IV
10	6.30	6.40	6.30	6.30
50	24.40	25.60	24.40	24.90
100	43.60	45.20	43.60	44.89
150	45.80	49.30	45.80	48.45
200	-	51.1	-	50.54

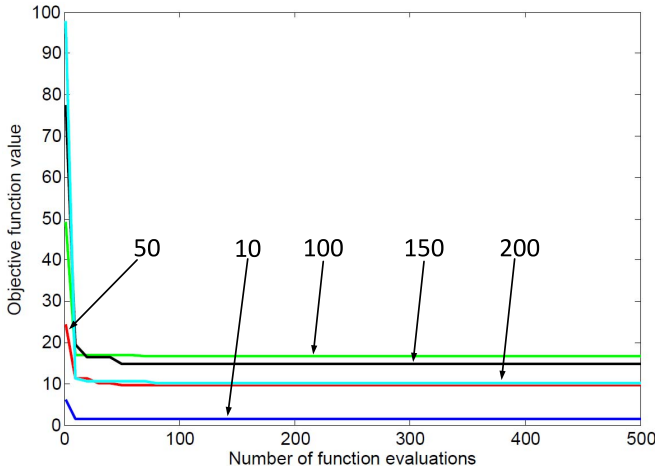


Fig. 3. Convergence of Method IV.

In the second test, we study how the computation time changes with the problem size. The setting is similar but we fix α to one for different values of n . We generate 10 feasible cases for n equal to 10, 50, 100, 150, and 200. Fig. 2 shows the average computation times of the four methods in the logarithmic scale. The corresponding objective function values normalized by computed minimums are also given for reference. All the computation times increase with n . Method I takes the longest computation time which also grows the fastest. Method III needs less time than Method IV when n is small. However, when n is larger than 100, Method III requires more time to compute the solution. In other words, the computation time of Method III grows faster than that of Method IV. Method II needs the shortest time but its computed solution quality is the worst. Although Methods I and III require relatively more time, their solutions are the best. Note that the results of Methods I and III for $n = 200$ are not

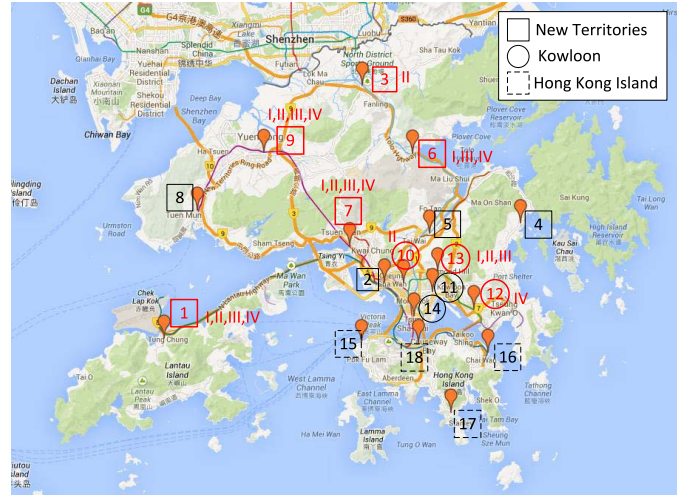


Fig. 4. Charging station distribution (adopted from [30]).

TABLE III
CHARGING STATION DATA (FROM THE 2006 CENSUS)

District	Population (k)	Density (/km ²)	Median monthly per capita (HK\$)	f_i
1	137.1	783	5659	1277.14
2	523.3	22421	4833	44.60
3	280.7	2055	5161	486.62
4	406.4	3135	6774	318.98
5	607.5	8842	6232	113.10
6	293.5	2156	5806	463.82
7	288.7	4679	6897	213.72
8	502.0	6057	5172	165.10
9	534.2	3858	4777	259.20
10	365.5	39095	4821	255.79
11	362.5	36178	6897	276.41
12	587.4	52123	4845	191.85
13	423.5	45540	4750	219.59
14	280.5	40136	6034	249.15
15	250.0	20102	9722	99.49
16	587.7	31664	7235	63.16
17	275.2	7083	6563	282.37
18	155.2	15788	10185	126.68

shown because they are not computable by YALMIP/CPLEX due to the out-of-memory problem. This implies the MILP approaches are not suitable for large problems.⁷ As before, Method IV always produces better solutions than Method II. The average numbers of charging stations constructed by the four methods are given in Table II. Moreover, we perform a series of tests to check the convergence of Method IV. We run CRO for some cases of different sizes used in the second test with duration up to 10000 FEs. Fig. 3 gives the convergence curves for particular cases; for clear illustration (of the performance at the beginning), we give the performance in first 500 FEs only. The results show that the algorithm converges very fast and can converge within 500 FEs in all the cases. Hence, it is concluded that our evaluation limit of 2000 for Method IV applied to all the three tests is sufficient.

⁷We run the simulations in MATLAB. The out-of-memory problem may happen at another n with a different combination of machine and platform. Here, we just demonstrate that Methods I and III are not scalable.

TABLE IV
SIMULATION RESULTS FOR THE HONG KONG CASE

α	D (km)	No. of Matched nodes	Objective function value (No. of stations constructed)						Computation time (s)			
			Method I	Method II	Method III	Method IV (mean)	Method IV (best)	Method IV (worst)	Method I	Method II	Method III	Method IV (mean)
1	30	4	19 181 (4)	19 181 (4)	19 181 (4)	19 181 (4)	19 181 (4)	19 181 (4)	16.81	5.77E-03	3.77	6.70
	35	2	9571 (2)	9571 (2)	9571 (2)	9571 (2)	9571 (2)	9571 (2)	17.14	5.46E-03	10.71	3.62
	40	2	9571 (2)	9571 (2)	9571 (2)	9571 (2)	9571 (2)	9571 (2)	17.31	5.86E-03	12.62	3.63
	45	2	9527 (2)	9527 (2)	9527 (2)	9527 (2)	9527 (2)	9527 (2)	17.89	5.75E-03	14.80	3.28
	50	2	9527 (2)	9527 (2)	9527 (2)	9527 (2)	9527 (2)	9527 (2)	17.85	5.56E-03	18.30	3.76
0.8	30	-	-	-	-	-	-	-	-	-	-	-
	35	1	23 768 (4)	30 001 (6)	23 768 (4)	23 768 (4)	23 768 (4)	23 768 (4)	17.17	6.88E-03	10.79	10.78
	40	1	11 718 (2)	14 348 (4)	11 718 (2)	11 718 (2)	11 718 (2)	11 718 (2)	17.45	5.31E-03	12.69	4.59
	45	2	9571 (2)	9571 (2)	9571 (2)	9571 (2)	9571 (2)	9571 (2)	17.76	5.30E-03	14.86	3.72
	50	2	9571 (2)	9571 (2)	9571 (2)	9571 (2)	9571 (2)	9571 (2)	17.86	5.59E-03	17.91	3.81
0.6	30	-	-	-	-	-	-	-	-	-	-	-
	35	-	-	-	-	-	-	-	-	-	-	-
	40	-	-	-	-	-	-	-	-	-	-	-
	45	4	27 889 (5)	30 065 (6)	27 889 (5)	27 912.7 (5)	27 889 (5)	27 984 (5)	17.62	7.55E-03	14.73	9.35
	50	4	19 181 (4)	19 181 (4)	19 181 (4)	19 181 (4)	19 181 (4)	19 181 (4)	17.79	4.83E-03	17.87	5.76

In the third test, we apply the problem to a real-world environment; we determine the locations for building charging stations in Hong Kong (HK). The HK Government plans to introduce more EVs (e.g., taxi) into the city and the construction of charging stations is one of the crucial steps in the plan. We can see how this can be realized through solving EVCSPP. HK is composed of three zones (New Territories, Kowloon, and HK Island) and each zone is further divided into districts. There are total 18 districts in HK. We select one location in each district for potential charging station construction. The location distribution is given in Fig. 4. The distance between each pair of locations is retrieved from the route connecting them suggested by Google Map [30]. We relate the location parameters to the district data obtained from [31]. We assign the population size to the demand F_i and the median monthly income per capita to the cost c_i . We set the capacity f_i inversely proportional to the density with some proportionality constants. The location parameter values are listed in Table III. We perform simulations with several combinations of D (30, 35, 40, 45, and 50 km) and α (1, 0.8, and 0.6) and the performance of the methods is given in Table IV where the best objective function values are bold and the numbers of stations constructed are put in brackets. The number of nodes matched in the solutions of the four methods (the best solutions for Method IV) for each feasible case is also provided. In Fig. 4, for the case with α and D equal to 0.6 and 45 km, respectively, those locations in red indicate that charging stations should be constructed according to the solutions computed by the four methods; the roman numbers beside a location reveal which methods have included that location in their solutions.⁸

Methods I and III always find the best solutions for all cases. Method IV can determine the best solutions in most cases while Method II can still achieve the best solutions in some cases. For computation time, on the average, Method II is the fastest, and then Method IV. Method III comes next and Method I takes the longest. In general, the computation times of Methods I, II, and IV decrease with D since \hat{G} becomes

denser with D and we need less effort to locate the solutions. But Method III does the opposite because nodes tend to have larger degrees with D . Thus, the number of MILPs needed to be solved increases together with the minimum degree of \hat{G} .

B. Discussion

From the simulations above, we can see that each method has its own characteristics and is suitable for different situations. Here, we try to compare them in terms of five different perspectives independently.

1) *Solution Quality*: If the adopted MILP solver can guarantee optimality,⁹ Methods I and III can obtain the best results. As Method II is embedded in Method IV, Method IV is always superior to Method II but they may not produce the optimal solutions, especially when the solution space is getting larger. They can be ranked as: I = III > IV > II.

2) *Computation Time*: Method II is of the simplest design and takes very limited amount of time to obtain a (usually sub-optimal) solution. Method I needs to apply MILP to all nodes of the problem while Method III requires only a subset. Hence, Method III is always faster than Method I. Method IV is a metaheuristic and we can terminate the algorithm when certain stopping criteria are satisfied (in our cases, we limit the computation time by setting an FE limit). As a whole, they can be ranked as: II > IV > III > I.

3) *Solvable Problem Size*: Solving MILP is the major building block of Methods I and III and it relies on the adopted solver. As most existing solvers can handle relatively small problems (in our case with MATLAB/YALMIP/CPLEX, the problem is only solvable with $n \leq 150$ in this paper). However, since the manipulating mechanisms of Methods II and IV are mainly about how to modify and evaluate temporary solutions, they are more resistive to the growing problem size. So they can be ranked as: II = IV > I = III.

4) *Algorithmic Nature*: All methods are deterministic except Method IV. In other words, we always come up with the identical result in different runs of the same problem instance. Method IV is probabilistic in nature. For each instance, we

⁸As the solutions of Method IV computed in different runs can be different, only its solution of a particular run is given in Fig. 4.

⁹Most MIP solvers can generate the optimal solutions when the problem size is small.

TABLE V
SOLUTION METHOD CHARACTERISTIC COMPARISON

	Method I	Method II	Method III	Method IV
Solution quality	✓✓✓	✓	✓✓✓	✓✓
Computation time	✓	✓✓✓✓	✓✓	✓✓✓
Problem size	✓	✓✓✓	✓	✓✓✓
algorithmic nature	Deterministic	Deterministic	Deterministic	Probabilistic
Prerequisite	None	None	Some	None

repeat the simulations several times to obtain its average performance.

5) *Prerequisite*: Recall that Method III is valid only when the condition imposed in Theorem 3 holds. Although this condition is very general and held in most practical situations, the other methods do not require it.

We summarize the characteristics of the four methods in Table V. Note that the above conclusions are drawn independently of each other from our observations on the simulation. When evaluating a particular method, we usually take several aspects into account simultaneously, e.g., correlating solution quality with computation time and problem size. However, we aim to give an extensive assessment and thus we appraise their individual abilities from one aspect to another. In general, each method has its own pros and cons and none is outstanding predominantly. In practice, we select the most suitable method according to our need.

As a final remark, we aim to show that the greedy algorithm can be considered as a component in a metaheuristic, whose performance is guaranteed to be better than that of Method II. Method IV can be interpreted in a broader sense; it is a greedy-algorithm embedded metaheuristic approach where we can replace CRO with any metaheuristic like Genetic Algorithm [32]. Further exploration of Method IV with other metaheuristics will be left for the future work.

VII. CONCLUSION

Gasoline is a heavily demanded natural resource and most is consumed on transportation. Transportation electrification can relieve our dependence on gasoline and tremendously reduce the amount of harmful gases released, which partially constitute global warming and worsen our health. In the 21st century, advancing EV technologies has become one of the keys to boost a nation's economy and maintain (and improve) people's quality of living. For long-term planning, modernizing our cities with EVs is of utmost importance. EVs will be integrated into the transportation system seamlessly and this will help make our cities "smart." To do this, we need to determine the best locations to construct charging stations in the city. In this paper, we focus on human factors rather than technological ones for charging station placement. An EV should always be able to access a charging station within its driving capacity anywhere in the city. Our contributions in this paper include: 1) formulating the problem; 2) identifying its properties; and 3) developing the corresponding solution methods. We formulate the problem as an optimization model, based on the charging station coverage and the convenience of drivers. We prove the problem NP-hard and propose four solution methods to tackle the problem. Each method has its own

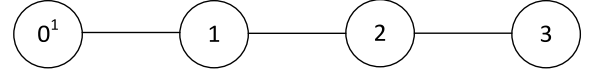


Fig. 5. Example of four nodes.

characteristics and is suitable for different situations depending on the requirements for solution quality, algorithmic efficiency, problem size, nature of the algorithm, and existence of system prerequisite.

APPENDIX ILLUSTRATIVE EXAMPLE FOR THE NETWORK FLOW MODEL

We make use of the example of \hat{G} given in Fig. 5 to illustrate the network flow model discussed in Section III-B.

Consider that nodes 1 and 2 have charging stations constructed and thus we have node 0^1 sends out two units of flow on $(0^1, 1)$ and hence $y_{01}^1 = 2$. Equation (3) indicates that the conversation of flow and thus we have $y_{12}^1 = 1$ as node 1 is a sink of one unit of flow. Similarly, we get $y_{23}^1 = 0$. In this way, we can ensure that the resultant locations of the charging stations (nodes 1 and 2 in this case) are connected.

Consider another case that nodes 1 and 3 are the locations of charging stations. So we have $x_1 = x_3 = 1$ and $x_2 = 0$. Equation (4) results in $y_{01}^1 = 2$. Node 2 is not a sink and (2) confines $y_{12}^1 = 0$. To balance the incoming and outgoing flows at node 2, (3) makes $y_{23}^1 = 0$. However, node 3 is a sink of one unit and when (3) is applied to node 3, we need to have $y_{23}^1 = 1$. This results in a contradiction and hence we cannot allow constructing charging stations at nodes 1 and 3 without node 2.

Therefore, connectivity of the charging station network can be enforced with the network flow model.

REFERENCES

- [1] A. Y. S. Lam, Y.-W. Leung, and X. Chu, "Electric vehicle charging station placement," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Vancouver, BC, Canada, Oct. 2013, pp. 510–515.
- [2] K. Lebeau, J. Van Mierlo, P. Lebeau, O. Mairesse, and C. Macharis, "Consumer attitudes towards battery electric vehicles: A large-scale survey," *Int. J. Elect. Hybrid Veh.*, vol. 5, no. 1, pp. 28–41, 2013.
- [3] S. Chen and L. Tong, "iEMS for large scale charging of electric vehicles: Architecture and optimal online scheduling," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Tainan City, Taiwan, Nov. 2012, pp. 629–634.
- [4] A. Y. S. Lam, L. Huang, A. Silva, and W. Saad, "A multi-layer market for vehicle-to-grid energy trading in the smart grid," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Orlando, FL, USA, Mar. 2012, pp. 85–90.
- [5] A. Y. S. Lam, K.-C. Leung, and V. O. K. Li, "Capacity management of vehicle-to-grid system for power regulation services," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Tainan, Taiwan, Nov. 2012, pp. 442–447.
- [6] J. J. Q. Yu, V. O. K. Li, and A. Y. S. Lam, "Optimal V2G scheduling of electric vehicles and unit commitment using chemical reaction optimization," in *Proc. IEEE Congr. Evol. Comput.*, Cancun, Mexico, Jun. 2013, pp. 392–399.
- [7] F. Guo, E. Inoa, W. Choi, and J. Wang, "Study on global optimization and control strategy development for a PHEV charging facility," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2431–2441, Jul. 2012.
- [8] M. Etezadi-Amoli, K. Choma, and J. Stefani, "Rapid-charge electric-vehicle stations," *IEEE Trans. Power Del.*, vol. 25, no. 3, pp. 1883–1887, Jul. 2010.

- [9] A. S. Masoum, S. Deilami, P. S. Moses, M. A. S. Masoum, and A. Abu-Siada, "Smart load management of plug-in electric vehicles in distribution and residential networks with charging stations for peak shaving and loss minimisation considering voltage regulation," *IET Gener. Transmiss. Distrib.*, vol. 5, no. 8, pp. 877–888, Aug. 2011.
- [10] Z. Liu, F. Wen, and G. Ledwich, "Optimal planning of electric-vehicle charging stations in distribution systems," *IEEE Trans. Power Del.*, vol. 28, no. 1, pp. 102–110, Jan. 2013.
- [11] Z. F. Liu, W. Zhang, X. Ji, and L. Ke, "Optimal planning of charging station for electric vehicle based on particle swarm optimization," in *Proc. IEEE Innov. Smart Grid Technol. Asia*, Tianjin, China, May 2012, pp. 1–5.
- [12] L. Jia, Z. Hu, Y. Song, and Z. Luo, "Optimal siting and sizing of electric vehicle charging stations," in *Proc. IEEE Int. Elect. Veh. Conf.*, Greenville, SC, USA, Mar. 2012, pp. 1–6.
- [13] E. Pashajavid and M. A. Golkar, "Optimal placement and sizing of plug in electric vehicles charging stations within distribution networks with high penetration of photovoltaic panels," *J. Renew. Sustain. Energy*, vol. 5, no. 5, pp. 101–118, Jan. 2013.
- [14] T. D. Chen, K. M. Kockelman, and M. Khan, "The electric vehicle charging station location problem: A parking-based assignment method for Seattle," in *Proc. 31st USAEE/IAEE North Amer. Conf.*, Austin, TX, USA, Nov. 2012, pp. 1–14.
- [15] S. Wang, R. Crosier, and Y. Chu, "Investigating the power architectures and circuit topologies for megawatt superfast electric vehicle charging stations with enhanced grid support functionality," in *Proc. IEEE Int. Elect. Veh. Conf.*, Greenville, SC, USA, Mar. 2012, pp. 1–8.
- [16] S. Khuller, A. Malekian, and J. Mestre, "To fill or not to fill: The gas station problem," *ACM Trans. Algorithm.*, vol. 7, no. 3, pp. 36:1–36:16, Jul. 2011.
- [17] Z. Drezner, *Facility Location: Applications and Theory*, 2nd ed. Berlin, Germany: Springer-Verlag, 2010.
- [18] R. Church and C. R. Velle, "The maximal covering location problem," *Papers Reg. Sci.*, vol. 32, no. 1, pp. 101–118, Jan. 1974.
- [19] D. Mayfield, "Site design for electric vehicle charging stations," *Sustain. Transport. Strategies*, Tech. Rep., Jul. 2012.
- [20] J. M. Conrad, C. P. Gomes, W.-J. van Hoeve, A. Sabharwal, and J. F. Suter, "Wildlife corridors as a connected subgraph problem," *J. Environ. Econ. Manag.*, vol. 63, no. 1, pp. 1–18, Jan. 2012.
- [21] S. Y. Hui, C. K. Lee, and F. F. Wu, "Electric springs—A new smart grid technology," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1552–1561, Sep. 2012.
- [22] B. Zhang, A. Y. S. Lam, A. Dominguez-Garcia, and D. Tse, "Optimal distributed voltage regulation in power distribution networks," *IEEE Trans. Power Syst.*, to be published.
- [23] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979.
- [24] A. Y. S. Lam and V. O. K. Li, "Chemical-reaction-inspired meta-heuristic for optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 381–399, Jun. 2010.
- [25] A. Y. S. Lam, V. O. K. Li, and J. Xu, "On the convergence of chemical reaction optimization for combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 605–620, Oct. 2013.
- [26] J. Xu, A. Y. S. Lam, and V. O. K. Li, "Chemical reaction optimization for task scheduling in grid computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 1624–1631, Oct. 2011.
- [27] A. Y. S. Lam, V. O. K. Li, and J. J. Q. Yu, "Power-controlled cognitive radio spectrum allocation with chemical reaction optimization," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3180–3190, Jul. 2013.
- [28] IBM ILOG CPLEX V12.1: User's manual for CPLEX, International Business Machines Corp., 2009.
- [29] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Symp. Comput. Aided Control Syst. Des.*, Taipei, Taiwan, Sep. 2004, pp. 284–289.
- [30] Hong Kong. (2013, Sep.). *Google Maps* [Online]. Available: <http://maps.google.com>
- [31] Wikipedia. (2013, Sep.). *Districts of Hong Kong* [Online]. Available: http://en.wikipedia.org/wiki/Districts_of_Hong_Kong
- [32] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.



Albert Y.S. Lam (S'03–M'10) received the B.Eng. (First Class Hons.) degree in information engineering, and the Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Hong Kong, in 2005 and 2010, respectively.

He is a Research Assistant Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, and was a Post-Doctoral Scholar with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA, from 2010 to 2012.

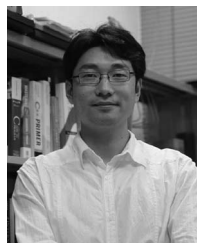
His current research interests include optimization theory and algorithms, evolutionary computation, smart grid, smart city planning, and Internet protocols and applications.

Dr. Lam is a Croucher Research Fellow.



Yiu-Wing Leung (M'92–SM'96) received the B.Sc. and the Ph.D. degrees in information engineering from the Chinese University of Hong Kong, Hong Kong, in 1989 and 1992, respectively.

He has been with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, and is currently a Full Professor. His current research interests include networking and Internet systems, and systems engineering and optimization. He has published over 80 journal papers in these areas, most of which were published in various IEEE journals.



Xiaowen Chu (M'01–SM'12) received the B.Eng. degree in computer science from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 2003.

He is currently an Associate Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include parallel and distributed computing, wireless networks, and cloud computing.